AIM-package : a modular standard for structuring patches in Max

Vincent Goudard
vincent@vincentgoudard.com

ABSTRACT

Designing a complete in-car audio experience requires solutions for rapid prototyping in a complex audio configuration, bringing together different areas of expertise ranging from sound-design and composition, down to hardware protection, with every conceivable layer of audio-engineering in-between, up to A-B comparisons setups for end-users perception evaluation in real demonstration vehicules. The AIM-package started as a request from the "Active Sound Experience" team at Volvo Cars Company [1] to meet such goals.

To this end, it was decided to develop a framework on top of Max [2] so that dedicated audio processing modules could be easily created, with the ability to store presets for various configurations, and to take advantage of Max's modular design to distribute the complexity of audio engineering among the various expert teams involved in the project.

The AIM-package was much designed after two older Max packages, namely *Jamoma Modular* — a part of the Jamoma project [1] started in 2006, and its current continuation in *libossia*, a part of the OSSIA project [2] [3] . Their main goal is well described in [1] : "to address concerns of sharing and exchanging Max patches in a modular system. This means creating a structured framework that does enforce consistency, readability, and standards for interoperability while not placing daunting restrictions on users." Beyond this goal, these two projects have a lot in common, with libossia being actively developed by a number of people previously involved in Jamoma. They both adopt a MVC design [4] , along with a number of conventions that contribute to their integration in the Max eco-system.

While the AIM-package is following a lot of these conventions, it was however decided to develop an alternative package —rather than just using libossia, for a number of reasons, that will be discussed here below and during the demo. Since the features of Jamoma have been described in [1], I will only stress a few differences and let the reader refer to the mentioned reference for the similarities.

[1] `https://www.volvocars.com`
[2] `https://cycling74.com`
[3] `http://www.jamoma.org` and `https://ossia.io`
[4] Model View Controler

Explicit model nesting
Contrary to Jamoma and libossia, models in AIM are explicitly bound to a parent model. This allows for a more flexible patcher organization, more consistent with the MVC design, and the possibility to dynamically rebind a model to another parent.

Typed modules
AIM modules are typed. This allows to share presets between all instances of modules of the same type. Namely, any preset created in a given model instance will readily be available to any other instance of that model, even if nested in another component.

Multichannel, multiplicity
The AIM-package is oriented towards multichannel processing [5] . To this end, it adopts a number of existing conventions to address MC objects in Max, and also supports wildcard and brace-expansion notation [6] to easily create or address multiplicities of nodes in the namespace tree.

Vanilla Max only
The AIM-package only relies on vanilla-Max objects. This generally proves to be more sustainable, as there is not C-coded external object, which compilation might break after a system update. The main drawback of course, is a worse performance and loading time, but these were acceptable enough for making this choice.

Acknowledgments

The development of the AIM-package was fully supported by Volvo Cars Company, which consider to release it as a free and open-source package for the community. For more information on the ASX project, please contact Jonatan Ewald [7] at Volvo Cars.

1. REFERENCES

[1] T. Place and T. Lossius, "Jamoma: A modular standard for structuring patches in max," in *ICMC*, 2006.

[2] J.-M. Celerier, "Authoring interactive media: a logical & temporal approach," Ph.D. dissertation, Bordeaux, 2018.

[5] ...as enabled with the introduction of MC.* objects in Max.
[6] `https://www.gnu.org/software/bash/manual/html_node/Brace-Expansion.html`
[7] jonatan.ewald@volvocars.com